In the claims:

1. (original) A method for reducing test data volume in the testing of logic products, comprising the steps of:

(a) generating original test vector data including care bits and non-care bits;

(b) filling said non-care bits with a repeated value to form a highly compressible test vector data set; and

(c) compressing said highly compressible test vector data set to form a compressed test vector data set.

2. (original) The method of claim 1, further comprising the steps of:

transmitting said compressed test vector data set to a test system; and

recovering the care bits of said original test vector data from said compressed vector data set, for loading into input latches of a tester in said test system.

3. (original) The method of claim 1, wherein said step (b) comprises:

generating a background vector data set; and

forming a differential vector data set by XORing said care bits with corresponding bits in said background vector data set.

4. (currently amended) The method of claim 3, wherein said XORing sets ~~a substantial portion~~ approximately half of said care bits to a value of 0 in said differential vector data set.

5. (original) The method of claim 3, further comprising the step of attaching a header to said differential vector data set, said header identifying an algorithm and seed used to generate said background vector data set, wherein said differential vector data set with attached header form said highly compressible test vector data set.

6. (currently amended) The method of claim 5, ~~wherein said recovering step comprises:~~ further comprising the step of

decompressing said compressed test vector data set;

extracting said differential vector data set and attached header;

reconstructing said background vector data set from said header; and

XORing said reconstructed background vector data set with said extracted differential vector data set to form a reconstructed test vector data set.

7. (original) The method of claim 6, wherein said reconstructed test vector data set comprises the care bits of the original test vector data, with the non-care bits having the values of the corresponding background vector data bits.

8. (original) The method of claim 3, wherein said background vector data set comprises a random distribution of bits having values of both "0" and "1".

9. (original) A method for reducing test data volume in the testing of logic products, comprising the steps of:

generating redundant test vectors from original test vector data; and

utilizing a repeat capability of a tester to load input latches of the tester with the redundant vectors.

10. (original) The method of claim 9, wherein said step of generating redundant test vectors comprises repeating a care bit value encountered in a first test vector of said original test vector data in non-care bits of neighboring test vectors.

11. (original) The method of claim 10, wherein said original test vector data comprises a matrix of test vectors arranged in rows and columns, and said care bit value is repeated in the same column for each row of said matrix, until a different care bit value is encountered.

12. (original) The method of claim 11, wherein when said different care bit value is encountered, said different care bit value is repeated in the same column for each row of said matrix, until another different care bit value is encountered.

13. (original) A computer-usable medium storing computer-executable instructions, said instructions when executed implementing a process for reducing test data volume in the testing of logic products, comprising the steps of:

(a) in original test vector data comprising care bits and non-care bits, filling said non-care bits with a repeated value to form a highly compressible test vector data set; and

(b) compressing said highly compressible test vector data set to form a compressed test vector data set.

14. (original) The computer-usable medium of claim 13, said process further comprising:

transmitting said compressed test vector data set to a test system; and

recovering the care bits of said original test vector data from said compressed vector data set, for loading into input latches of a tester.

15. (original) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for generating redundant test vectors from original test vector data, by repeating a care bit value encountered in a first test vector of said original test vector data in non-care bits of neighboring test vectors.

16. (original) The program storage device of claim 15, wherein said original test vector data comprises a matrix of test vectors arranged in rows and columns, and said care bit value is repeated in the same column for each row of said matrix, until a different care bit value is encountered.

17. (original) A method comprising:

(i) forming a compressed test data set by setting care bits and non-care bits in original test data to a repeated value and compressing said original test data;

(ii) downloading said compressed test data set to a testing system for logic products;

(iii) decompressing said compressed test data set to form a decompressed test data set; and

(iv) loading input latches in a logic product with said decompressed data set.

18. (currently amended) The method of claim ~~18~~ 17, wherein said step (i) comprises XORing said care bits with corresponding bits of a background vector data set.

19. (original) The method of claim 18, wherein said background vector data set has a random distribution of bits having values of 0 and 1.

20. (currently amended) The method of claim 19, wherein said XORing sets ~~a substantial portion~~ approximately half of said care bits to a value of 0.

21. (original) A method for testing logic products comprising:

applying a first testing technique to said logic products during a first testing period, said first testing technique comprising loading input latches of a tester for testing said logic products with test vectors comprising a random distribution of bits; and

applying a second testing technique to said logic products during a second testing period following said first testing period, said second testing technique comprising loading said input latches by repeating test vectors of a minimum set of test vectors obtained by repeating a last care bit in neighboring non-care bit positions.

22. (original) The method of claim 21, wherein said first testing period is approximately 10% of the test patterns testing approximately 95% of the faults, and said second testing period is approximately 90% of the test patterns testing approximately 5% of the faults.

23. (currently amended) The method of claim 21, wherein said first testing period ~~consists of~~ comprises a plurality of tests ~~sufficient for achieving~~ that achieve a certain fault coverage threshold that is lower than the total fault coverage objective for the full testing period, and said second testing period consists of additional tests ~~sufficient to raise~~ raising the fault coverage to the total fault coverage objective for the full testing period.

24 (original) The method of claim 23 wherein said second test period would contain several more tests than said first period.